

Figure 1

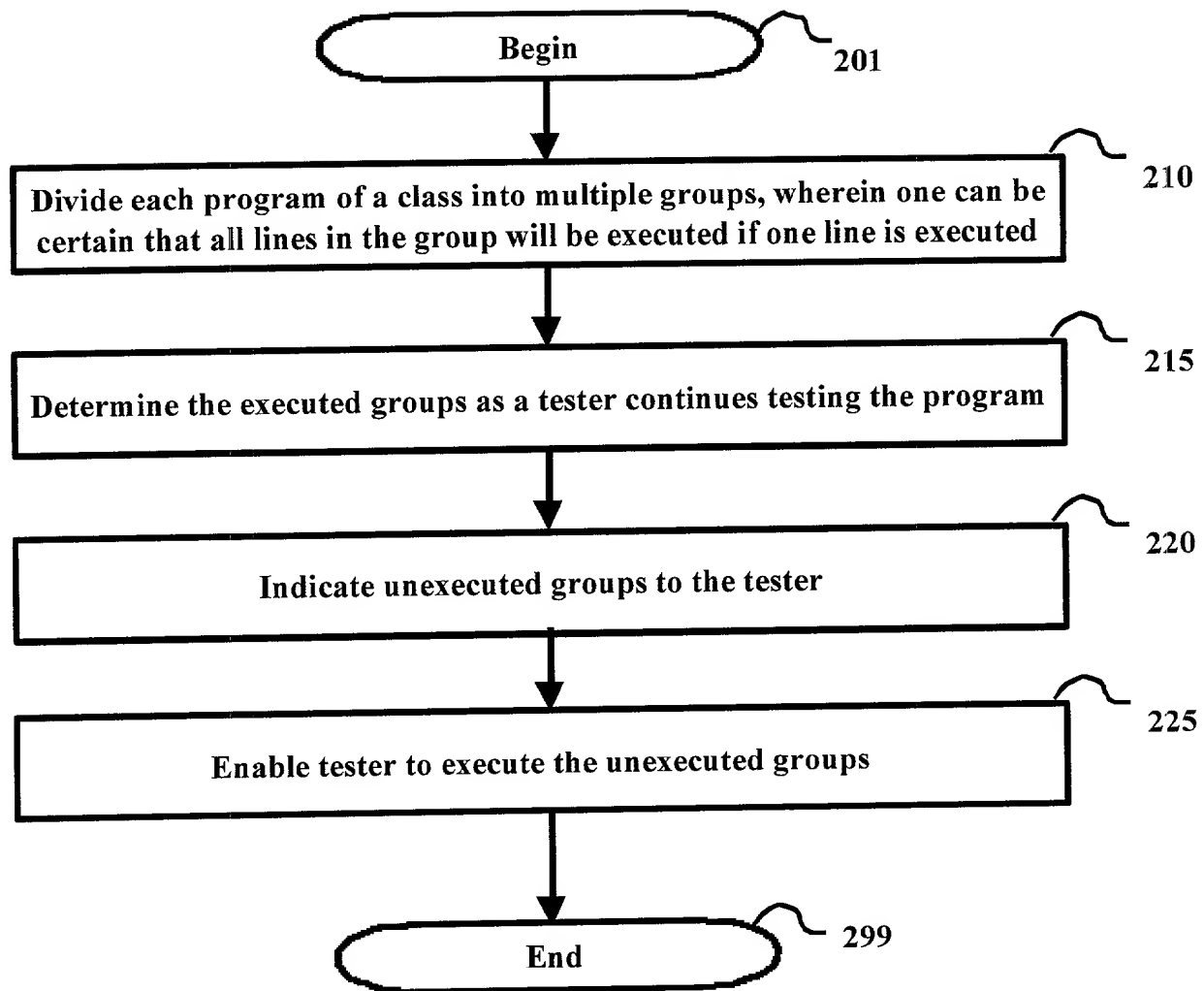


Figure 2

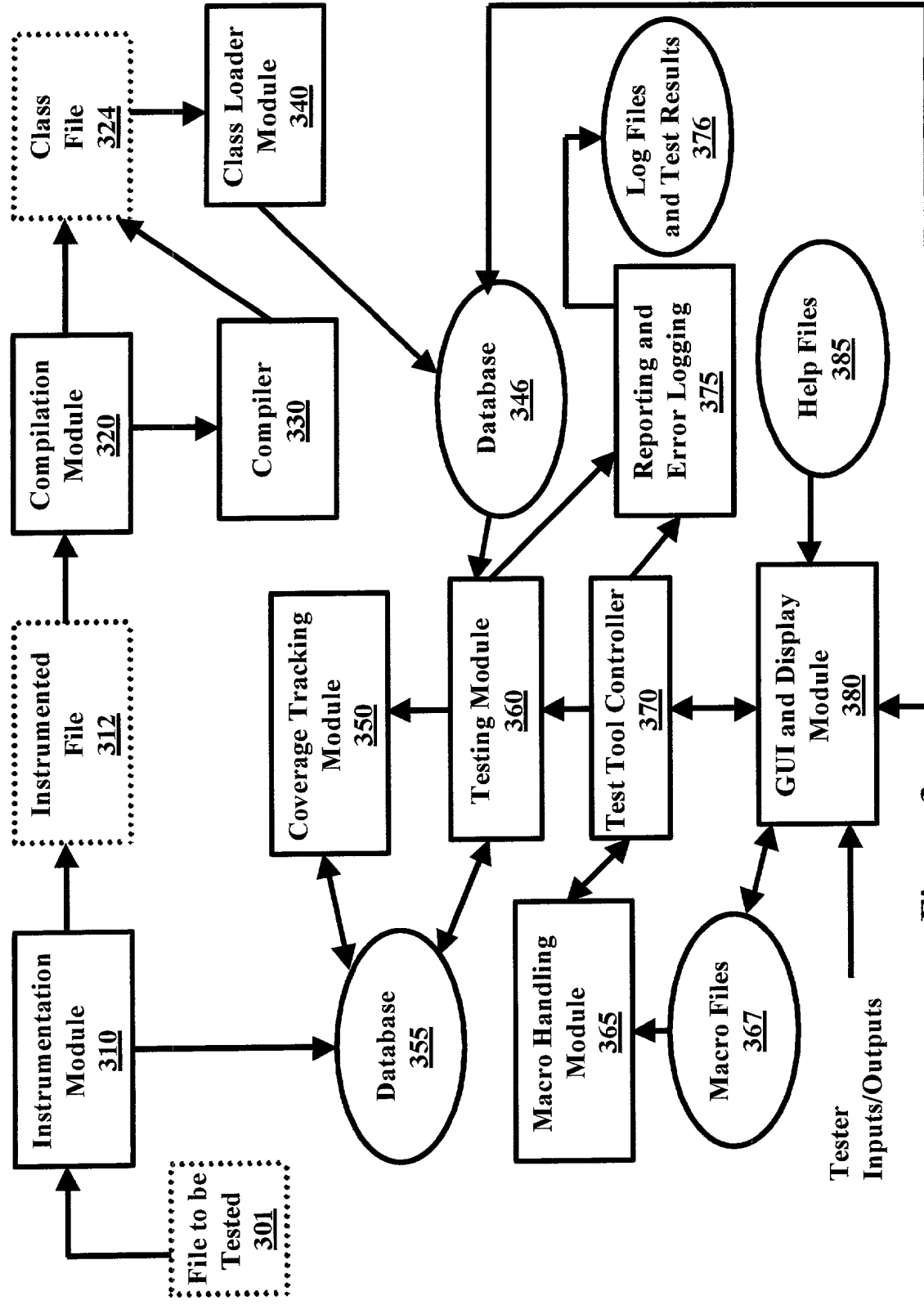


Figure 3

Attorney Docket No. : JP920000411US1

```

1      public void addString(String inStr) {
2          /* The location where the string needs to be set */
3          int setPos = -1;
4          /* Check the size to see if the next set needs to be allocated */
5          if( list.size() > 0 && (
6              (String)(list.elementAt(list.size()-1))).equals(emptyString) ) {
7              /* We still have empty space, get the first empty slot */
8              int totalRec = list.size();
9              for(int index=totalRec-1;index >= 0;index--) {
10                 if(!((String)list.elementAt(index)).equals(emptyString)) {
11                     break;
12                 }
13                 else {
14                     setPos = index;
15                 }
16             }
17         }
18         else {
19             /* We have run out of space, allocate the nest set */
20             int size = list.size();
21             for(int index=0;index<numSetVal;index++) {
22                 list.addElement(emptyString);
23             }
24             setPos = size;
25         }

26         /* Add the element */
27         list.setElementAt(inStr,setPos);
28     }

```

Figure 4A

```

1      public void addString(String inStr) {
2          /* INSTRUMENTATION Group B11 - BEGIN */
3          Wiz_Tracer.trackExecution("WizStringList",11);
4          /* INSTRUMENTATION GROUP B11 - END */
5          int setPos = -1;
6          if( list.size() > 0 && (
7      (String)(list.elementAt(list.size()-1))).equals(emptyString) )
8      { /*INSTRUMENTATION GROUP B12 - BEGIN */
9          Wiz_Tracer.trackExecution("WizStringList",12);
10         /* INSTRUMENTATION GROUP B12 - END */
11         int totalRec = list.size();
12         for(int index=totalRec-1;index >= 0;index--) {
13             /* INSTRUMENTATION GROUP B13 - BEGIN */
14             Wiz_Tracer.trackExecution("WizStringList",13);
15             /* INSTRUMENTATION GROUP B13 - END */
16             if(!((String)list.elementAt(index)).equals(emptyString)) {
17                 /* INSTRUMENTATION GROUP B14 - BEGIN */
18                 Wiz_Tracer.trackExecution("WizStringList",14);
19                 /* INSTRUMENTATION GROUP B14 - END */
20                 break;
21             } /* <if GROUP(B14) ends at line : 134> */

```

Figure 4B

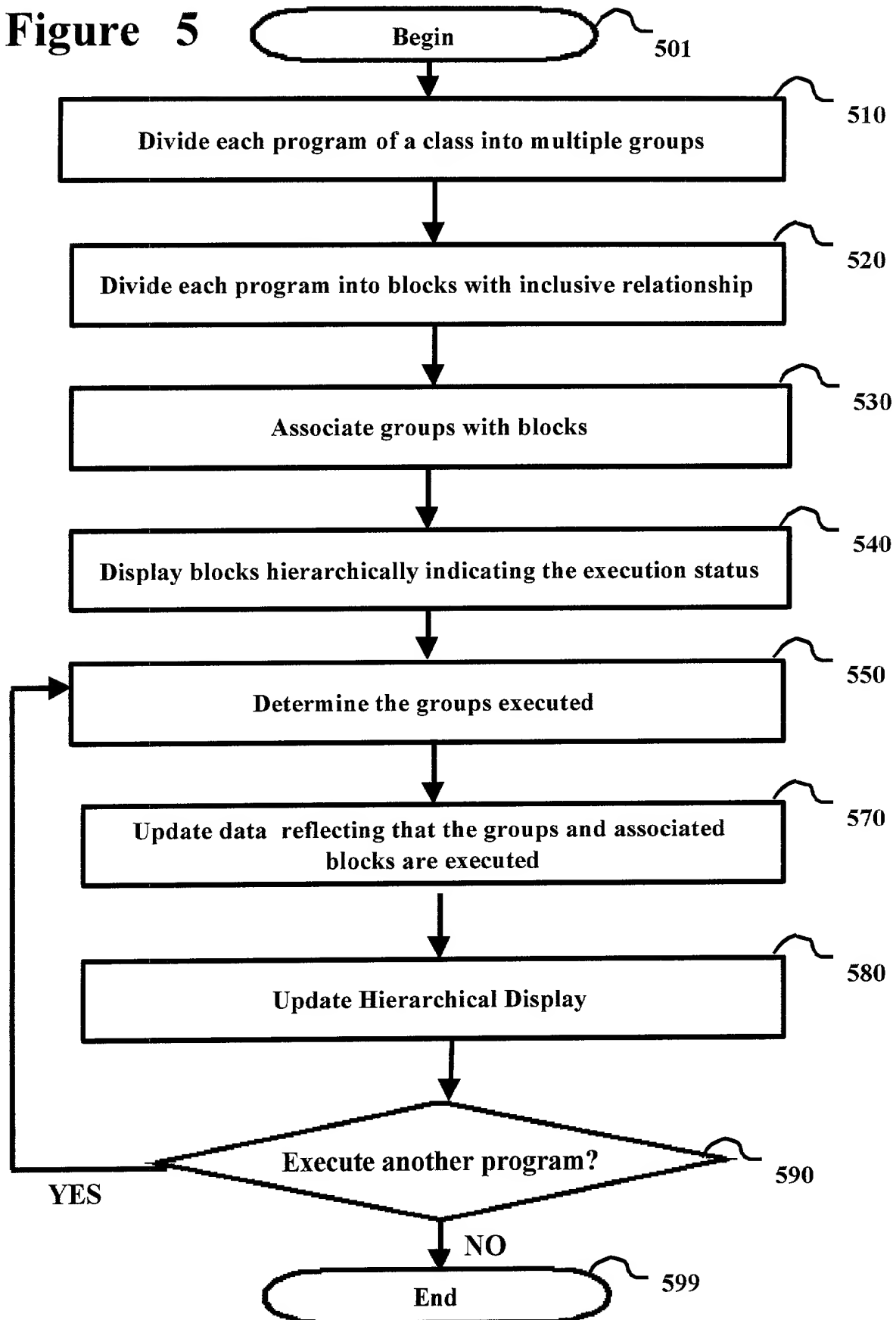
```

22             else { /* <else GROUP(B15) begins at line : 135> */
23                 /* INSTRUMENTATION GROUP B15 - BEGIN */
24                 Wiz_Tracer.trackExecution("WizStringList",15);
25                 /* INSTRUMENTATION GROUP B15 - END */
26                 setPos = index;
27     } /* <else block(B15) ends at line : 138> */
28         /* <break(B16) begins at line : 139> */
29         /* INSTRUMENTATION GROUP B16 - BEGIN */
30         Wiz_Tracer.trackExecution("WizStringList",16);
31         /* INSTRUMENTATION GROUP B16 - END */
32     } /* <for block(B13) ends at line : 139> */
33 } /* <if block(B12) ends at line : 141> */
34 else { /* <else block(B17) begins at line : 142> */
35     /* INSTRUMENTATION GROUP B17 - BEGIN */
36     Wiz_Tracer.trackExecution("WizStringList",17);
37     /* INSTRUMENTATION GROUP B17 - END */
38     int size = list.size();
39     for(int index=0;index<numSetVal;index++) { /* <for
40         block(B18) begins at line : 146> */
41         /* INSTRUMENTATION GROUP B18 - BEGIN */
42         Wiz_Tracer.trackExecution("WizStringList",18);
43         /* INSTRUMENTATION GROUP B18 - END */
44         list.addElement(emptyString);
45     } /* <for block(B18) ends at line : 149> */
46     setPos = size;
47 } /* <else block(B17) ends at line : 152> */
48 list.setElementAt(inStr,setPos);
49 } /* <method block - addString(B11) ends at line : 156> */

```

Figure 4C

Figure 5



CONFIDENTIAL

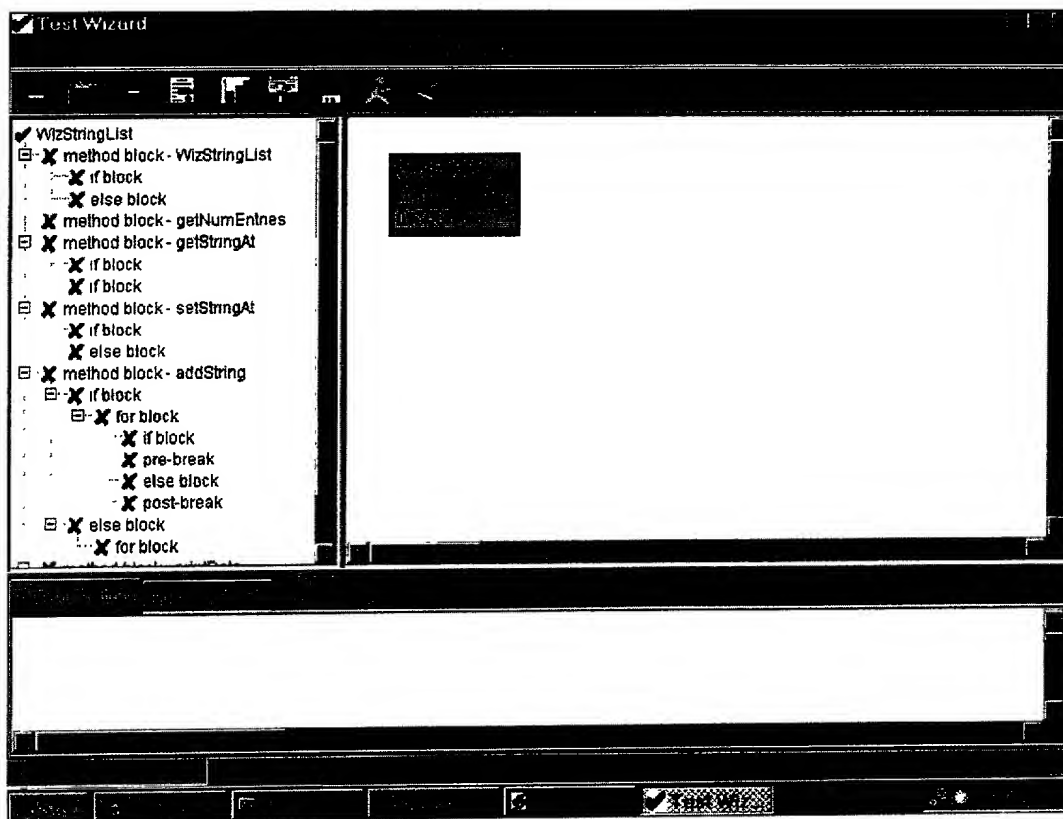


Figure 6

[illegible]

Attorney Docket No.: JP920000411US1

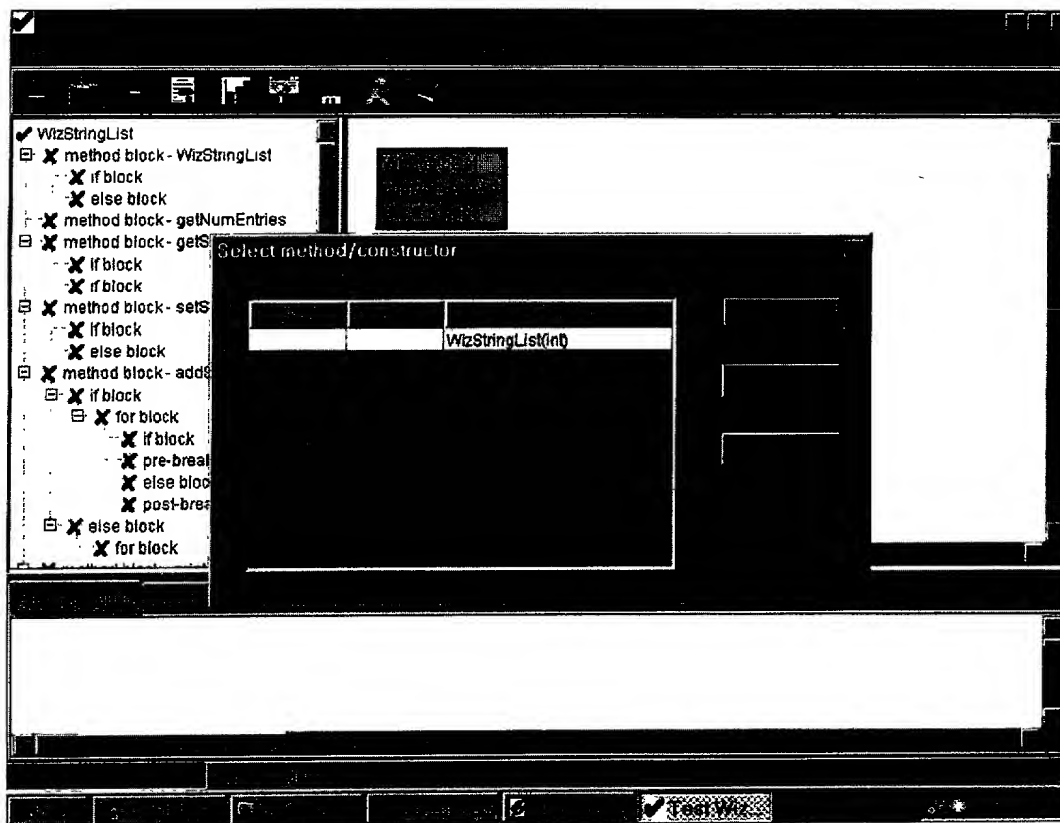


Figure 7B

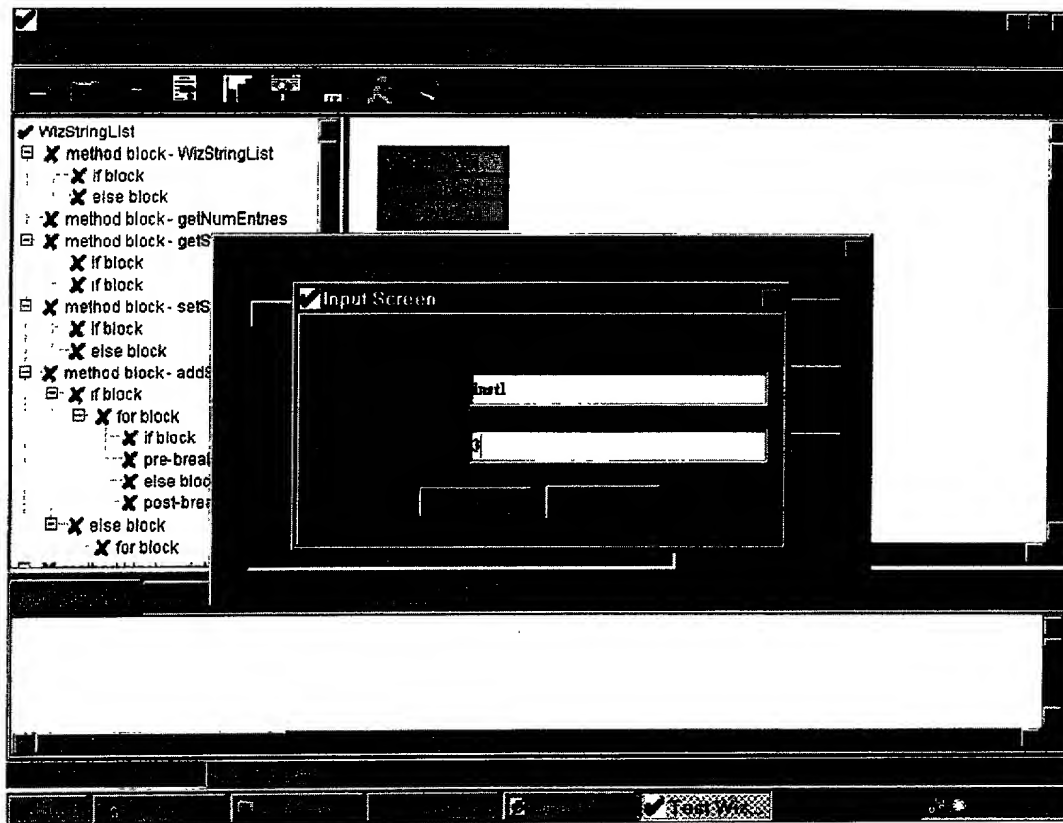


Figure 7C

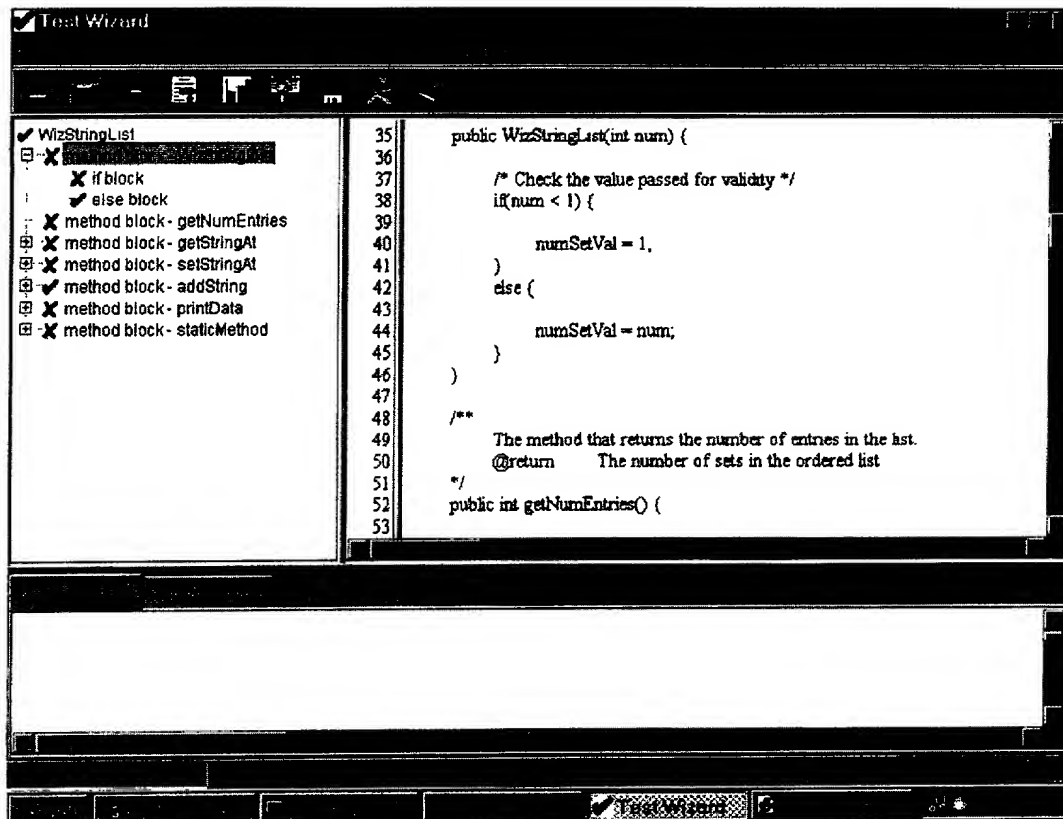


Figure 8

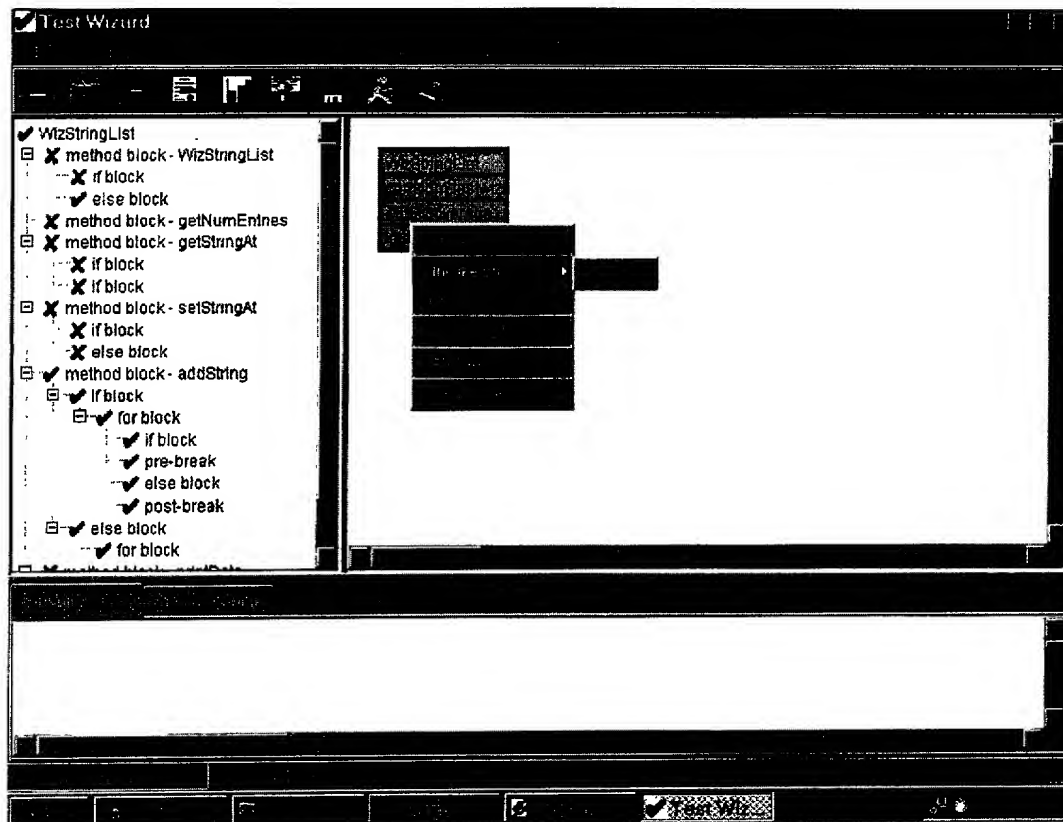


Figure 9A

The screenshot shows the NetBeans IDE with the 'WizStringList' class selected in the left pane. The 'Select method/constructor' dialog is open, displaying a table of methods and constructors for the selected class.

Method/Constructor	Return Type	Parameters
WizStringList	String	getStringAt(int, int)
WizStringList	boolean	setStringAt(int, int, String)
WizStringList	int	getNumEntries()
WizStringList	void	addString(String)
WizStringList	void	printData()
WizStringList	void	staticMethod(WizStringList)

Figure 9B

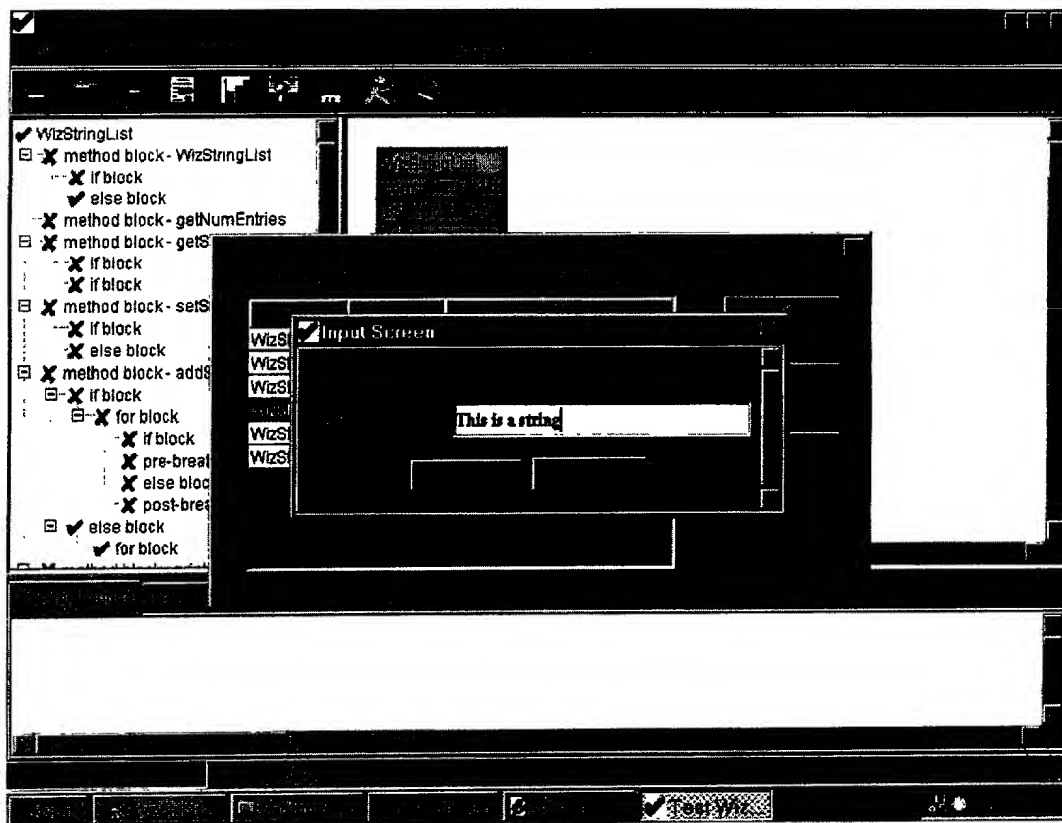


Figure 9C

The screenshot shows the NetBeans IDE interface. On the left, the Project Explorer displays a package named 'com.muhimbi' containing a class 'WizStringList'. The class has several methods, each with a checkbox for 'if block', 'else block', 'pre-break', 'post-break', and 'for block'. The 'Return value of method' window is open, showing two text areas. The top text area contains the text 'This is a string (String)' and the bottom text area contains the text 'Method returned void'. The window has 'Return' and 'Save As...' buttons at the bottom. The IDE's status bar at the bottom shows 'Run' and 'Save' buttons.

Figure 9D

1041630-0923260

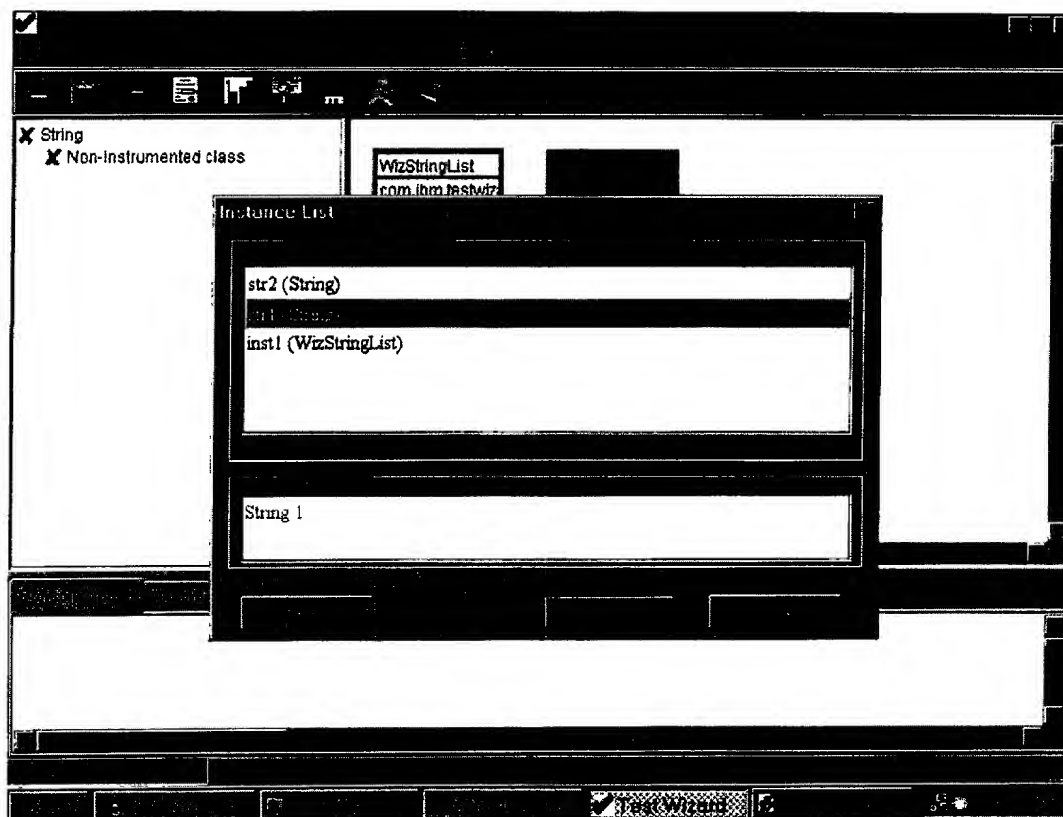


Figure 10

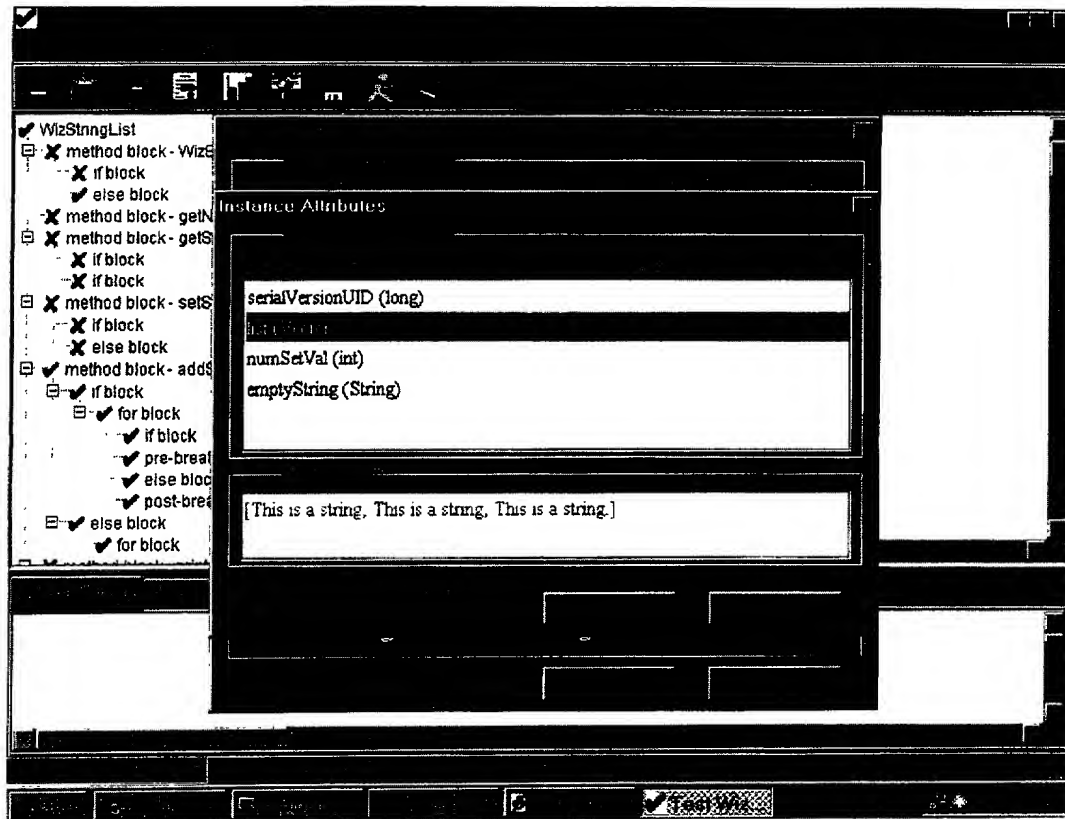


Figure 11A

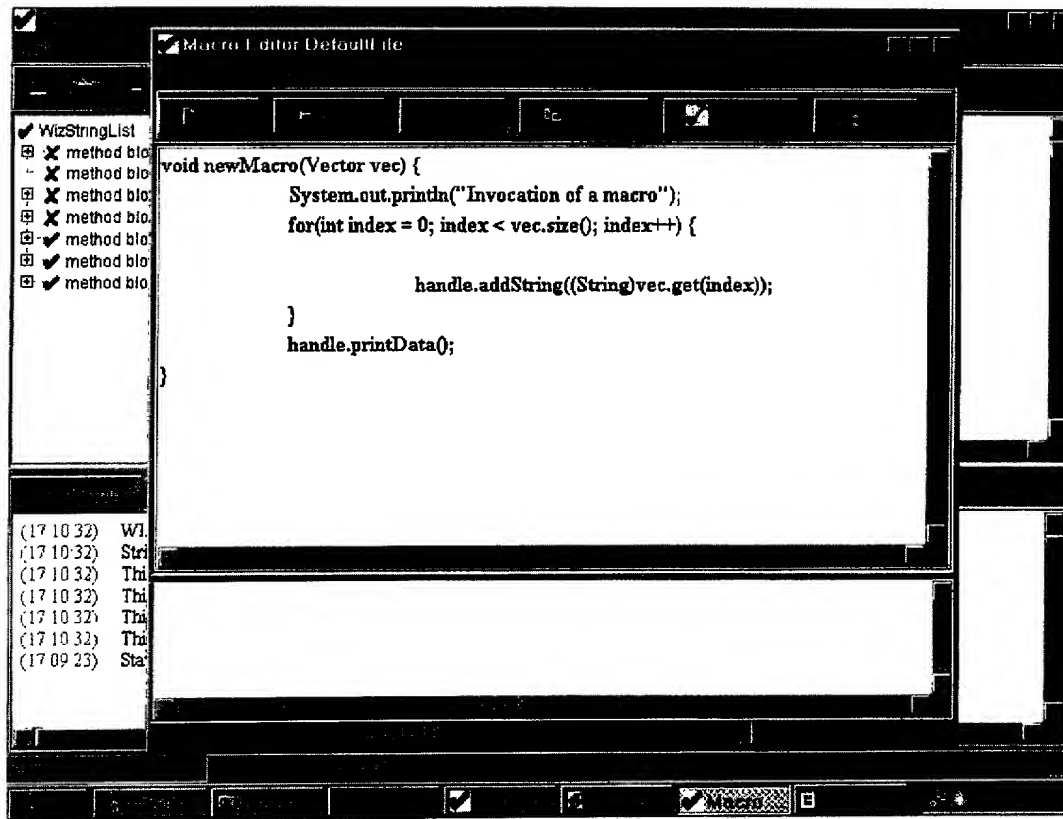


Figure 11B

FOR Filing - 09/09/2000

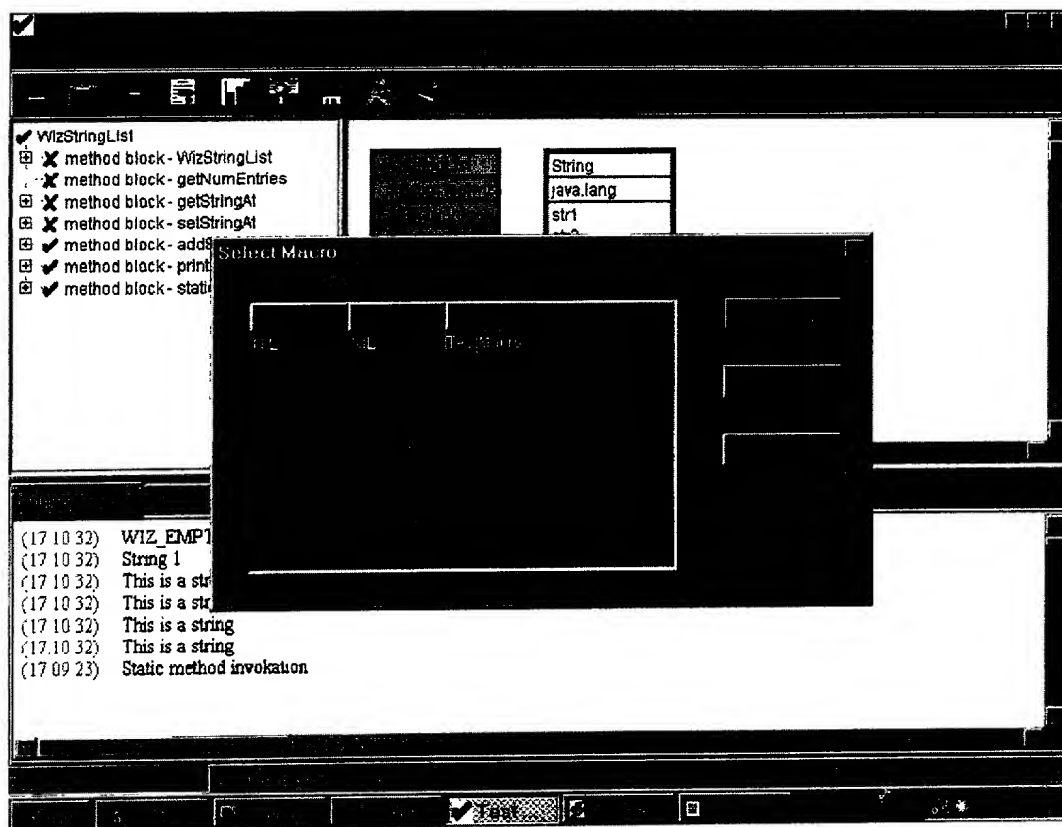


Figure 11C